

> Importación de datos

[] ↪ 1 celda oculta

✓ 1. Análisis de facturación

```
Precio_tienda_1 = sum(tienda['Precio'])  
print(Precio_tienda_1)
```

↪ 1150880400.0

```
Precio_tienda_2 = sum(tienda2['Precio'])  
print(Precio_tienda_2)
```

↪ 1116343500.0

```
Precio_tienda_3 = sum(tienda3['Precio'])  
print(Precio_tienda_3)
```

↪ 1098019600.0

```
Precio_tienda_4 = sum(tienda4['Precio'])  
print(Precio_tienda_4)
```

↪ 1038375700.0

```
Lista_total_preciosxtienda = [Precio_tienda_1, Precio_tienda_2, Precio_tienda_3, Precio_t  
print(Lista_total_preciosxtienda)
```

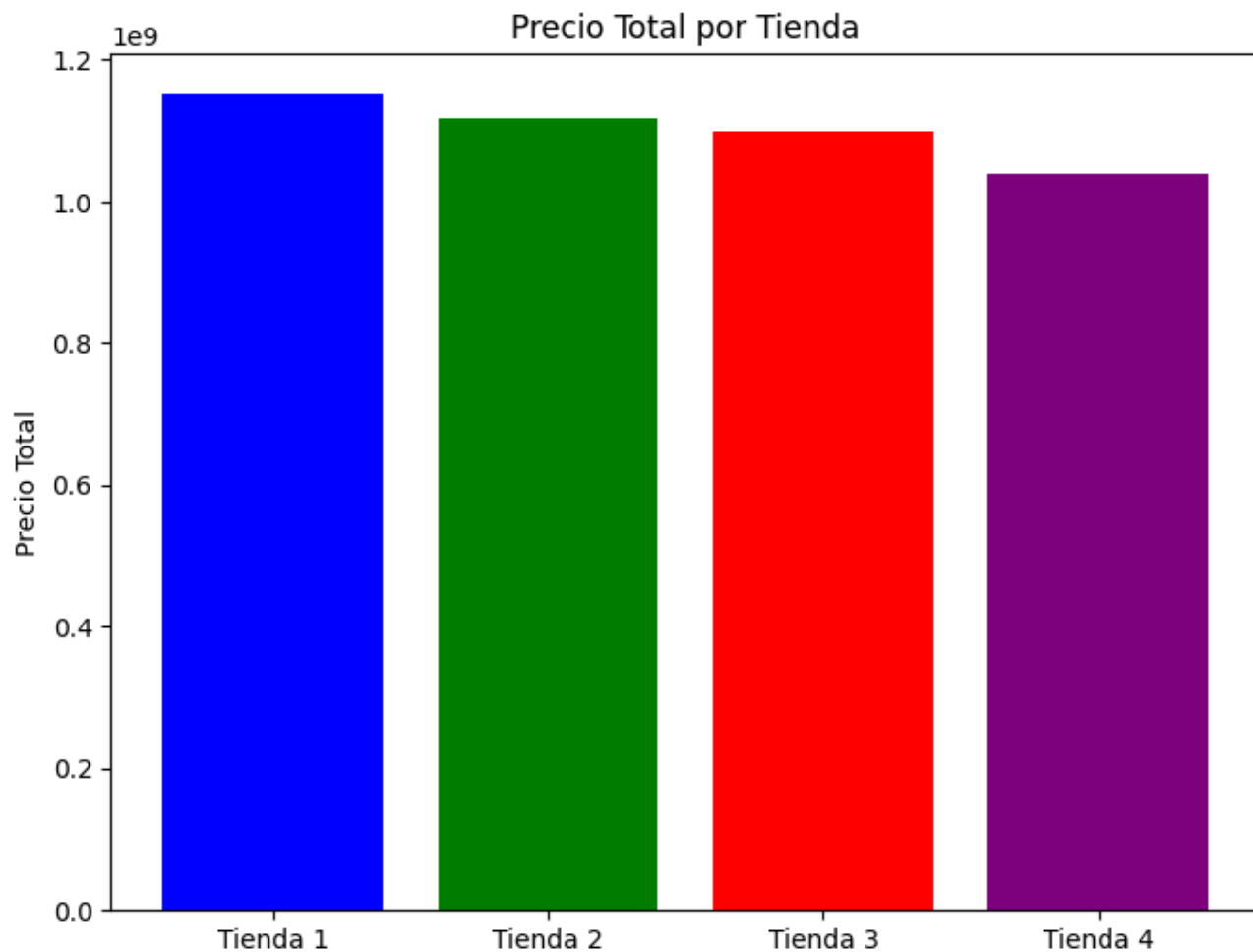
```
#Precio_total_tiendas = sum(Lista_total_preciosxtienda)  
#print(Lista_total_preciosxtienda)
```

```
Precio_total_tiendas = Precio_tienda_1+Precio_tienda_2+Precio_tienda_3+Precio_tienda_4  
print(Precio_total_tiendas)
```

↪ [1150880400.0, 1116343500.0, 1098019600.0, 1038375700.0]
4403619200.0

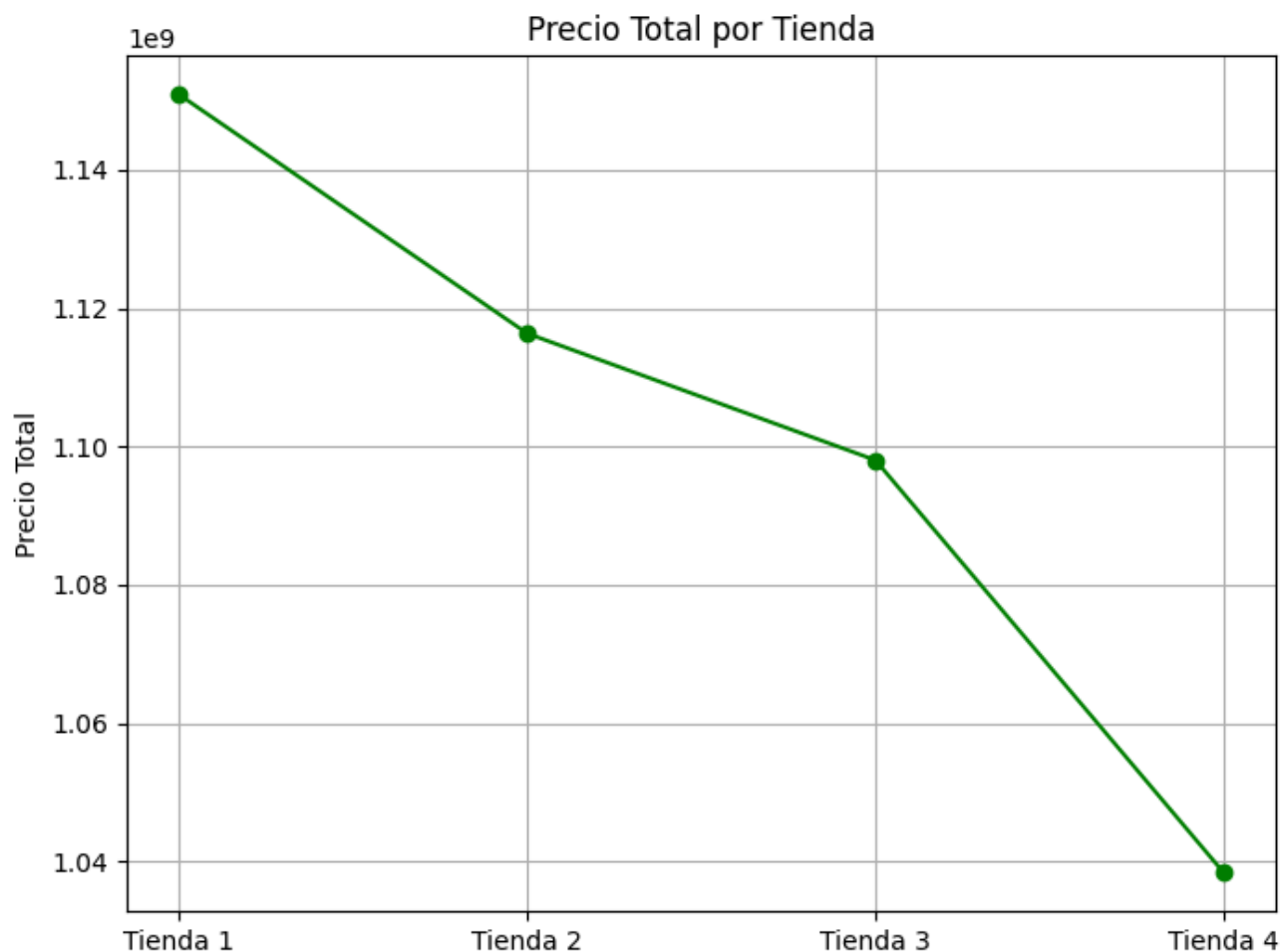
```
import matplotlib.pyplot as plt  
tiendas = ['Tienda 1', 'Tienda 2', 'Tienda 3', 'Tienda 4']
```

```
plt.figure(figsize=(8, 6))  
plt.bar(tiendas, Lista_total_preciosxtienda, color=['blue', 'green', 'red', 'purple'])  
plt.ylabel('Precio Total')  
plt.title('Precio Total por Tienda')  
plt.show()
```



```
import matplotlib.pyplot as plt
tiendas = ['Tienda 1', 'Tienda 2', 'Tienda 3', 'Tienda 4']

plt.figure(figsize=(8, 6))
plt.plot(tiendas, Lista_total_preciosxtienda, marker='o', linestyle='-', color='green')
plt.ylabel('Precio Total')
plt.title('Precio Total por Tienda')
plt.grid(True)
plt.show()
```



✓ 2. Ventas por categoría

```
categoria = tienda.groupby('Categoría del Producto')
print(categoria)
```

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7c5a051c5850>

```
import matplotlib.pyplot as plt
tiendas_data = {
    'Tienda 1': tienda,
    'Tienda 2': tienda2,
    'Tienda 3': tienda3,
    'Tienda 4': tienda4
}
```

```
for nombre_tienda, df_tienda in tiendas_data.items():
    ventas_por_categoria = df_tienda['Categoría del Producto'].value_counts()
    print(f"\nVentas por Categoría en {nombre_tienda}:")
    print(ventas_por_categoria)
```

```
# Opcional: Mostrar las categorías más populares
print(f"\nCategorías más populares en {nombre_tienda}:")
```

```
print(ventas_por_categoria.head()) # Muestra las 5 categorías más vendidas

plt.figure(figsize=(10, 6))
ventas_por_categoria.plot(kind='bar')
plt.title(f'Ventas por Categoría en {nombre_tienda}')
plt.xlabel('Categoría del Producto')
plt.ylabel('Cantidad de Productos Vendidos')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Ventas por Categoría en Tienda 1:

Categoría del Producto

Muebles	465
Electrónicos	448
Juguetes	324
Electrodomésticos	312
Deportes y diversión	284
Instrumentos musicales	182
Libros	173
Artículos para el hogar	171

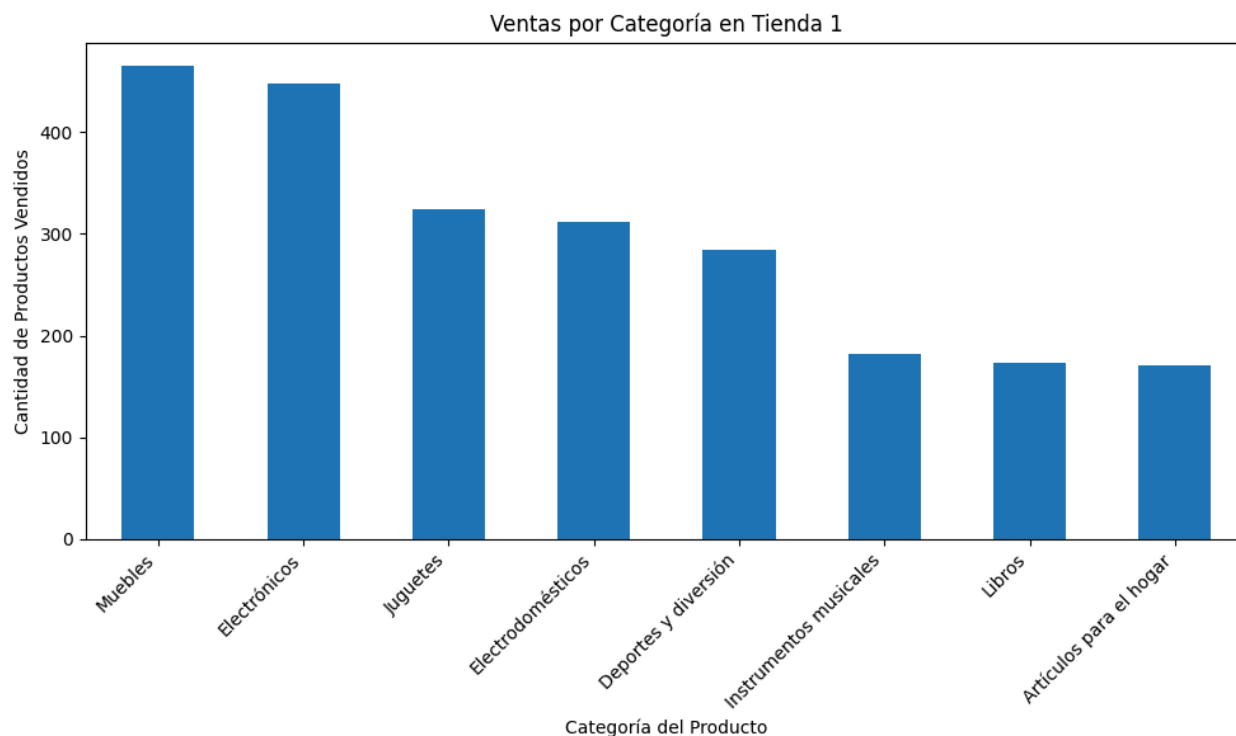
Name: count, dtype: int64

Categorías más populares en Tienda 1:

Categoría del Producto

Muebles	465
Electrónicos	448
Juguetes	324
Electrodomésticos	312
Deportes y diversión	284

Name: count, dtype: int64



Ventas por Categoría en Tienda 2:

Categoría del Producto

Muebles	442
Electrónicos	422
Juguetes	313
Electrodomésticos	305
Deportes y diversión	275
Instrumentos musicales	224
Libros	197
Artículos para el hogar	181

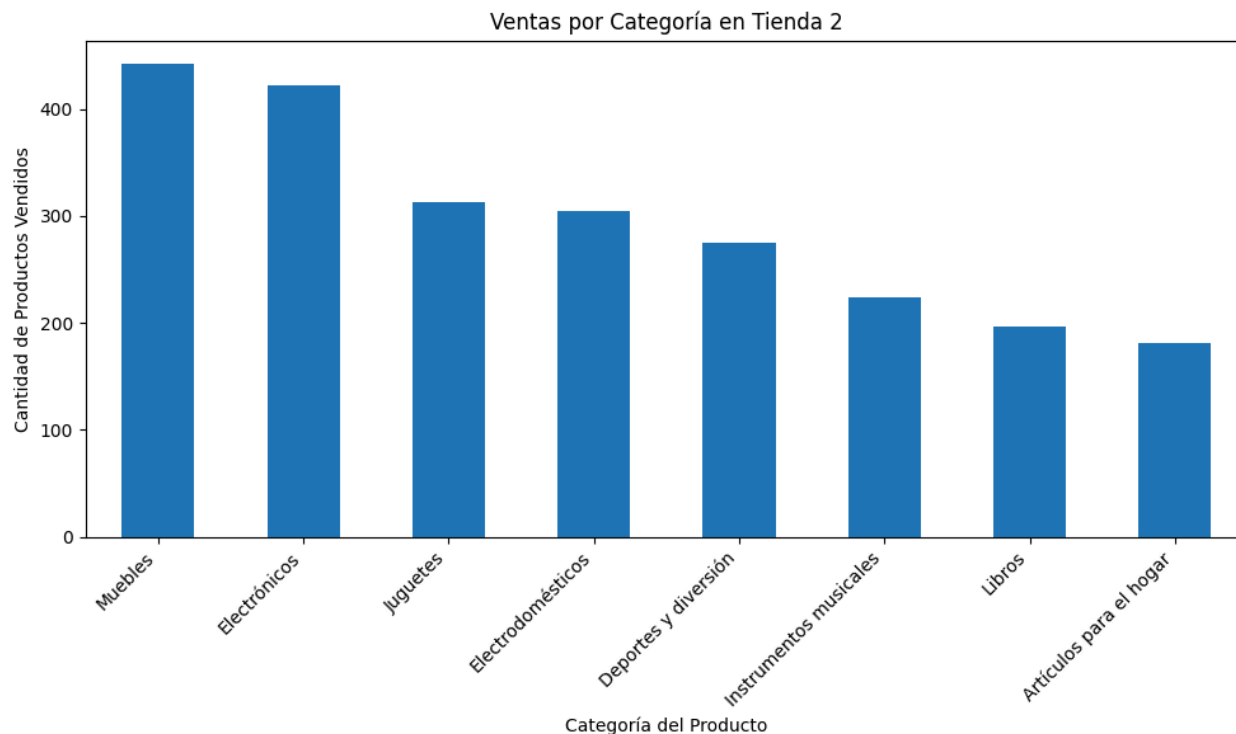
Name: count, dtype: int64

Categorías más populares en Tienda 2:

Categoría del Producto

Muebles	442
Electrónicos	422

```
Juguetes          315
Electrodomésticos 305
Deportes y diversión 275
Name: count, dtype: int64
```



Ventas por Categoría en Tienda 3:

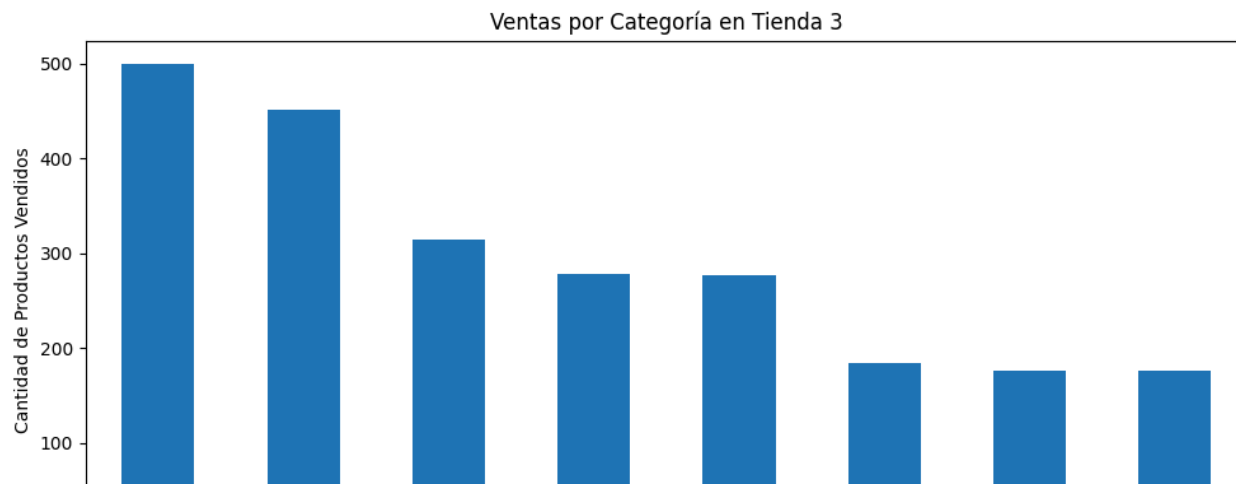
Categoría del Producto

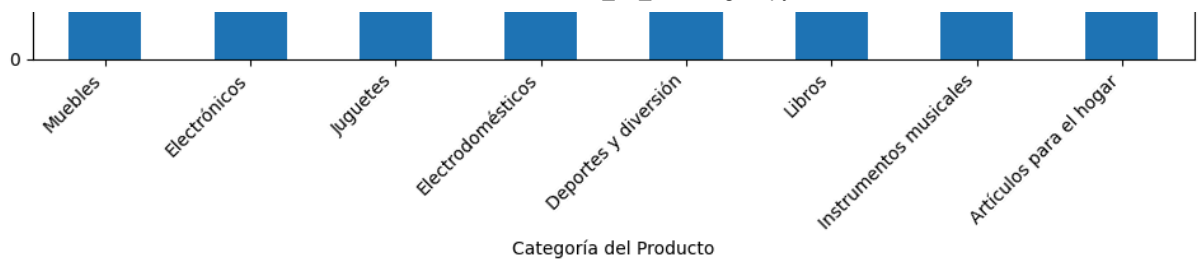
```
Muebles          499
Electrónicos      451
Juguetes          315
Electrodomésticos 278
Deportes y diversión 277
Libros            185
Instrumentos musicales 177
Artículos para el hogar 177
Name: count, dtype: int64
```

Categorías más populares en Tienda 3:

Categoría del Producto

```
Muebles          499
Electrónicos      451
Juguetes          315
Electrodomésticos 278
Deportes y diversión 277
Name: count, dtype: int64
```





Ventas por Categoría en Tienda 4:

Categoría del Producto

Muebles	480
Electrónicos	451
Juguetes	338
Deportes y diversión	277
Electrodomésticos	254
Artículos para el hogar	201
Libros	187
Instrumentos musicales	170

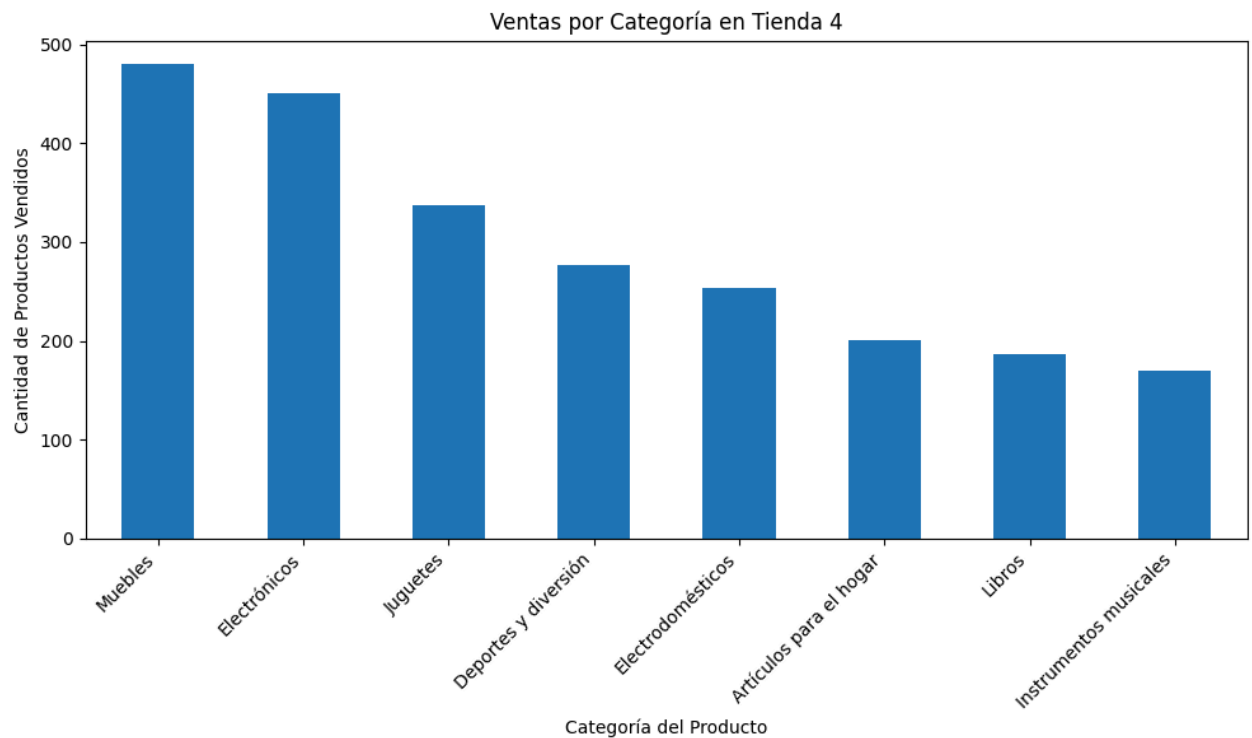
Name: count, dtype: int64

Categorías más populares en Tienda 4:

Categoría del Producto

Muebles	480
Electrónicos	451
Juguetes	338
Deportes y diversión	277
Electrodomésticos	254

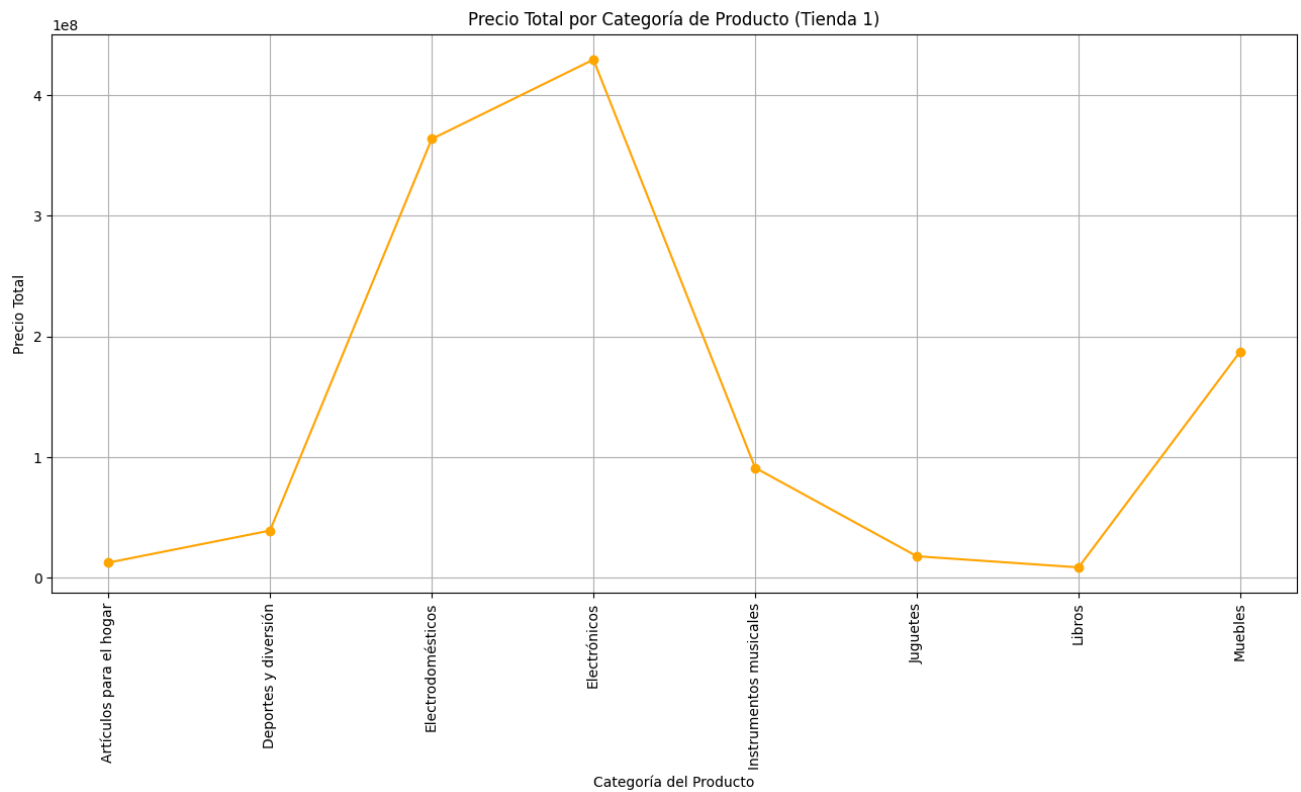
Name: count, dtype: int64




```
Precio_categoria = categoria['Precio'].sum()  
print(Precio_categoria)
```

```
→ Categoría del Producto  
Artículos para el hogar      12698400.0  
Deportes y diversión         39290000.0  
Electrodomésticos           363685200.0  
Electrónicos                 429493500.0  
Instrumentos musicales       91299000.0  
Juguetes                     17995700.0  
Libros                       8784900.0  
Muebles                      187633700.0  
Name: Precio, dtype: float64
```

```
import matplotlib.pyplot as plt  
plt.figure(figsize=(13, 8)) # Aumenta el tamaño de la figura  
plt.plot(Precio_categoria.index, Precio_categoria.values, marker='o', linestyle='-', color='b')  
plt.xlabel('Categoría del Producto')  
plt.ylabel('Precio Total')  
plt.title('Precio Total por Categoría de Producto (Tienda 1)')  
plt.xticks(rotation=90) # Rota las etiquetas del eje x para mejor legibilidad  
plt.grid(True)  
plt.tight_layout()  
plt.show()
```



```
# Función para analizar ventas por categoría en una tienda
def analyze_category_sales(df, store_name):
    category_sales = df.groupby('Categoría del Producto').size().reset_index(name='Cantidad')
    print(f"Ventas por Categoría en {store_name}:")
    print(category_sales.sort_values(by='Cantidad Vendida', ascending=False))
    print("-" * 30)

# Analizar ventas por categoría para cada tienda
analyze_category_sales(tienda, 'Tienda 1')
analyze_category_sales(tienda2, 'Tienda 2')
```

```
analyze_category_sales(tienda3, 'Tienda 3')
analyze_category_sales(tienda4, 'Tienda 4')
```

```
➡ Ventas por Categoría en Tienda 1:
  Categoría del Producto  Cantidad Vendida
7                Muebles             465
3            Electrónicos             448
5                Juguetes             324
2        Electrodomésticos             312
1    Deportes y diversión             284
4    Instrumentos musicales             182
6                Libros              173
0  Artículos para el hogar             171
```

```
-----
Ventas por Categoría en Tienda 2:
  Categoría del Producto  Cantidad Vendida
7                Muebles             442
3            Electrónicos             422
5                Juguetes             313
2        Electrodomésticos             305
1    Deportes y diversión             275
4    Instrumentos musicales             224
6                Libros              197
0  Artículos para el hogar             181
```

```
-----
Ventas por Categoría en Tienda 3:
  Categoría del Producto  Cantidad Vendida
7                Muebles             499
3            Electrónicos             451
5                Juguetes             315
2        Electrodomésticos             278
1    Deportes y diversión             277
6                Libros              185
0  Artículos para el hogar             177
4    Instrumentos musicales             177
```

```
-----
Ventas por Categoría en Tienda 4:
  Categoría del Producto  Cantidad Vendida
7                Muebles             480
3            Electrónicos             451
5                Juguetes             338
1    Deportes y diversión             277
2        Electrodomésticos             254
0  Artículos para el hogar             201
6                Libros              187
4    Instrumentos musicales             170
```

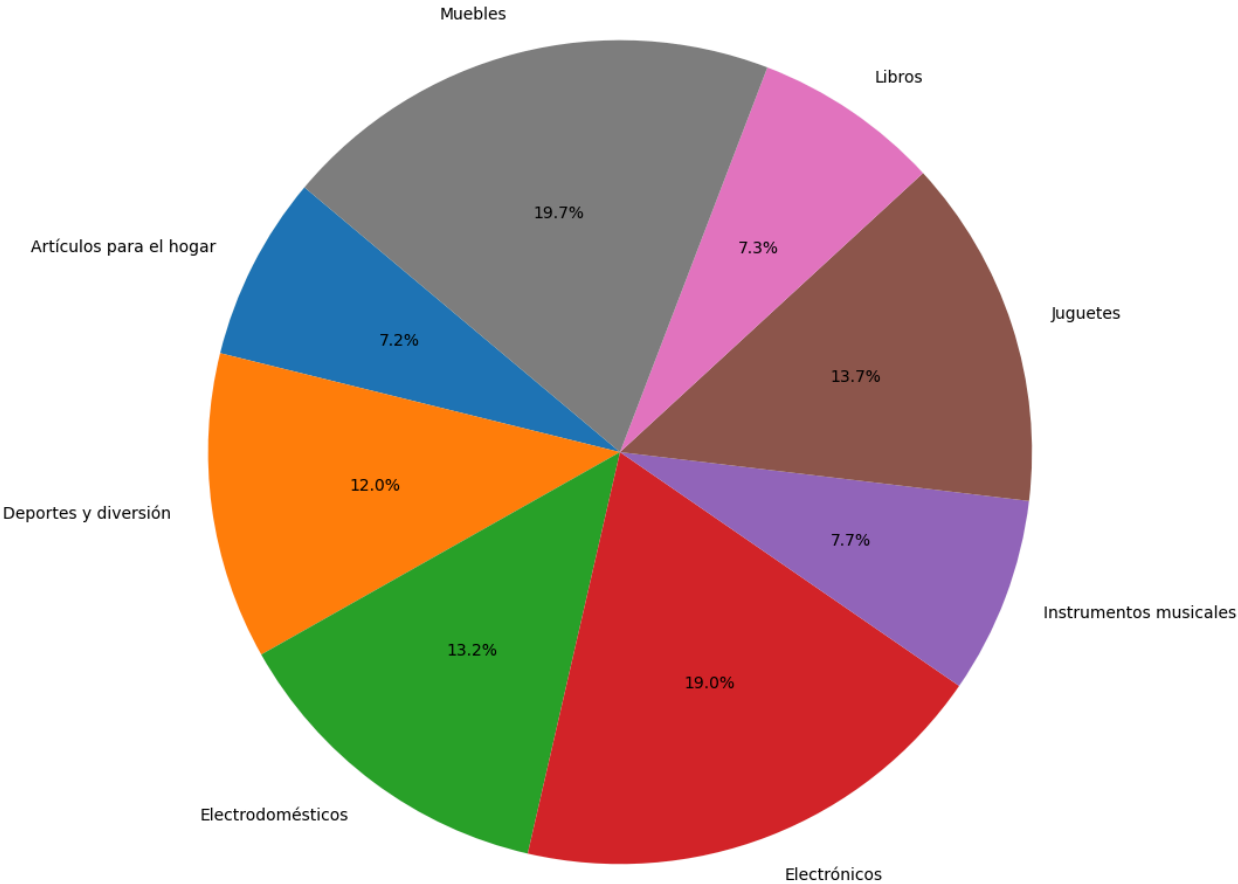
```
import matplotlib.pyplot as plt
def analyze_category_sales_pie(df, store_name):
    category_sales = df.groupby('Categoría del Producto').size().reset_index(name='Cantidad')

    plt.figure(figsize=(10, 10)) # Aumenta el tamaño de la figura
    plt.pie(category_sales['Cantidad'], labels=category_sales['Categoría del Producto'])
    plt.title(f'Distribución de Ventas por Categoría en {store_name}', fontsize=16)
    plt.axis('equal') # Asegura que el gráfico de pastel sea un círculo
    plt.show()
    print("-" * 30)
```

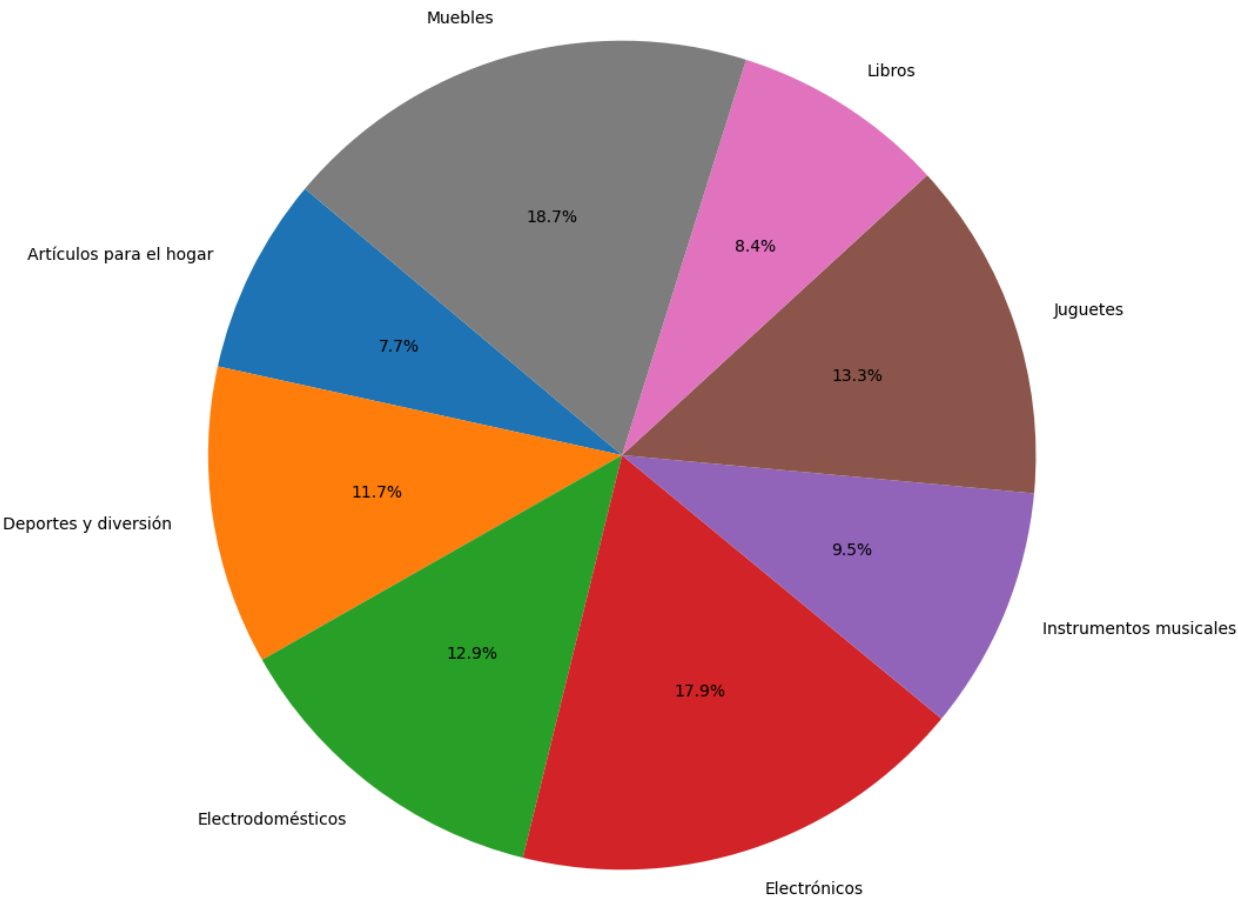
```
# Analizar y generar gráficos de pastel de ventas por categoría para cada tienda
analyze_category_sales_pie(tienda, 'Tienda 1')
analyze_category_sales_pie(tienda2, 'Tienda 2')
analyze_category_sales_pie(tienda3, 'Tienda 3')
analyze_category_sales_pie(tienda4, 'Tienda 4')
```



Distribución de Ventas por Categoría en Tienda 1

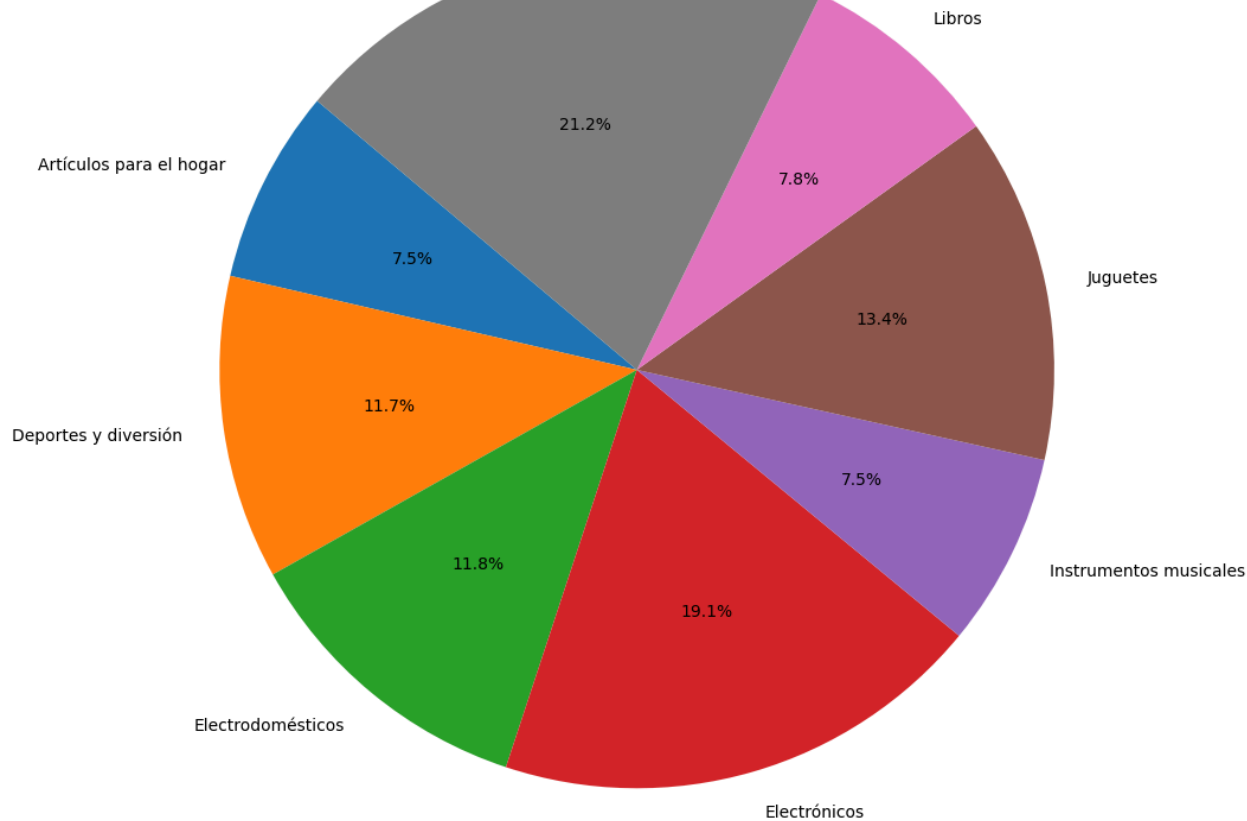


Distribución de Ventas por Categoría en Tienda 2

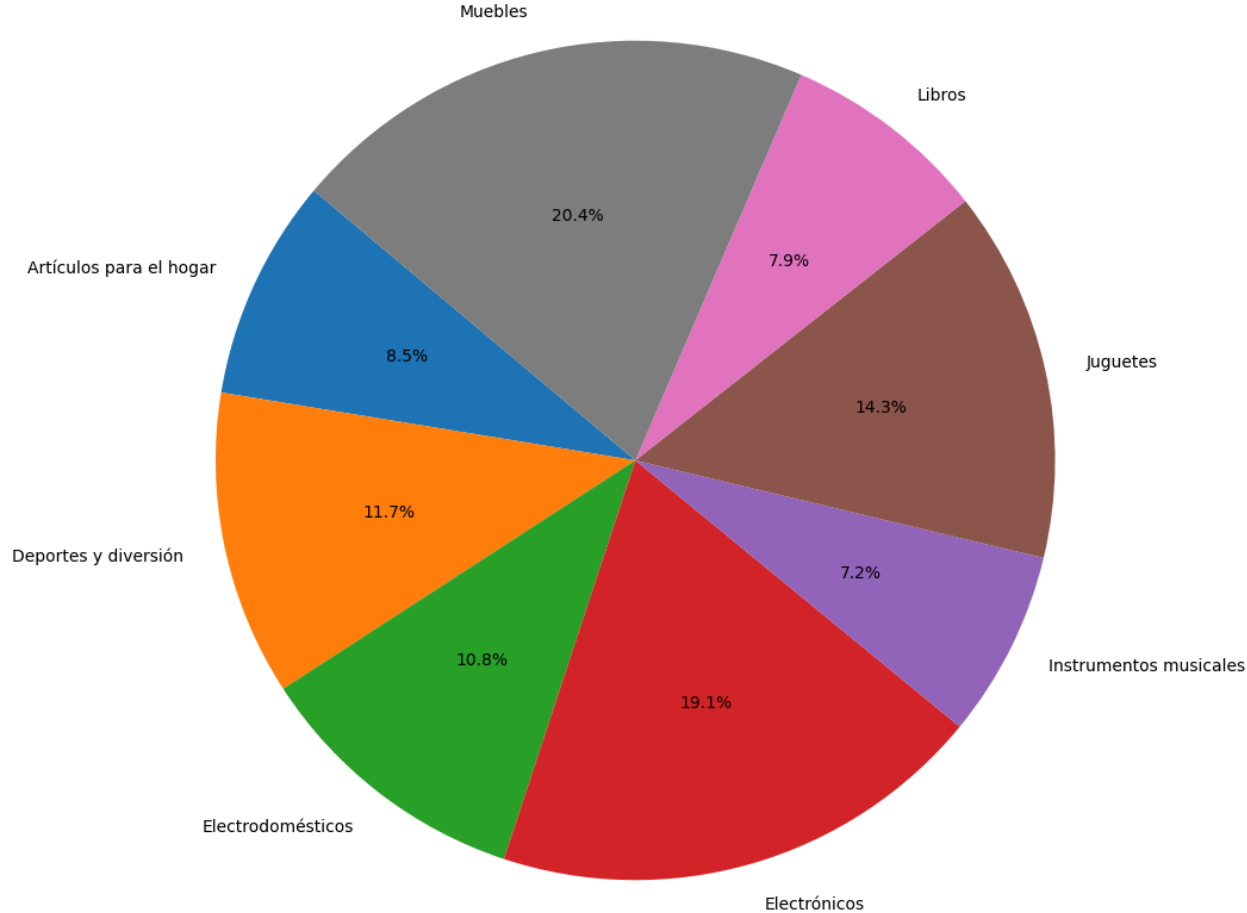


Distribución de Ventas por Categoría en Tienda 3

Muebles



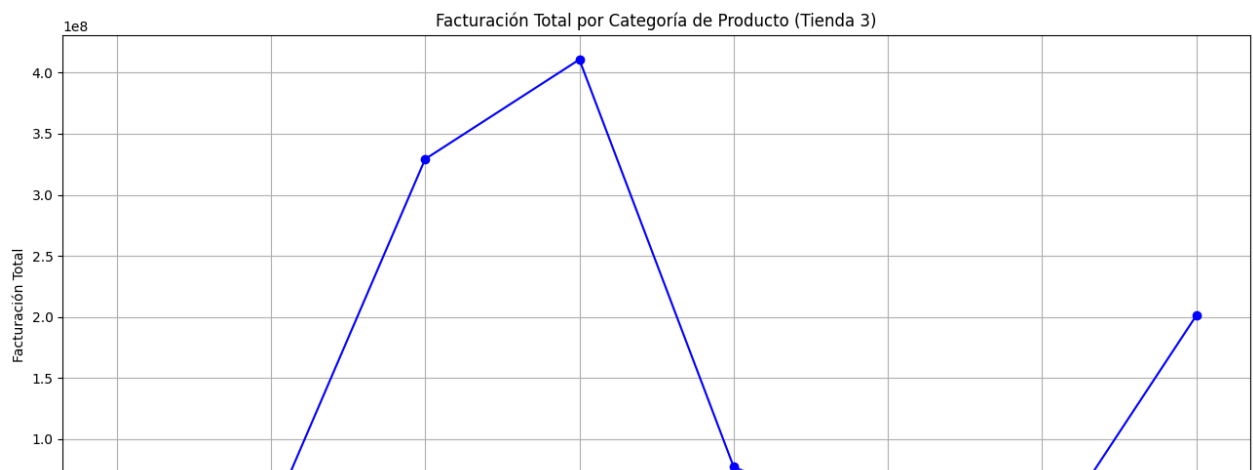
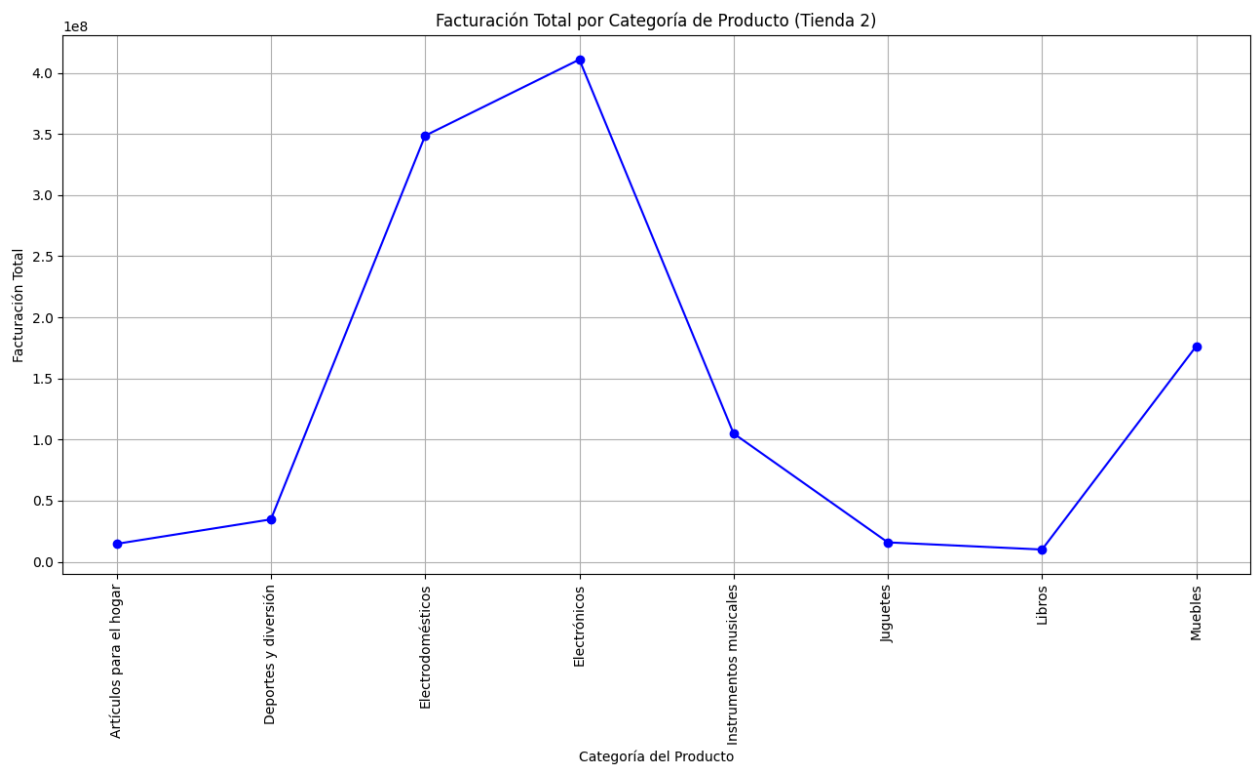
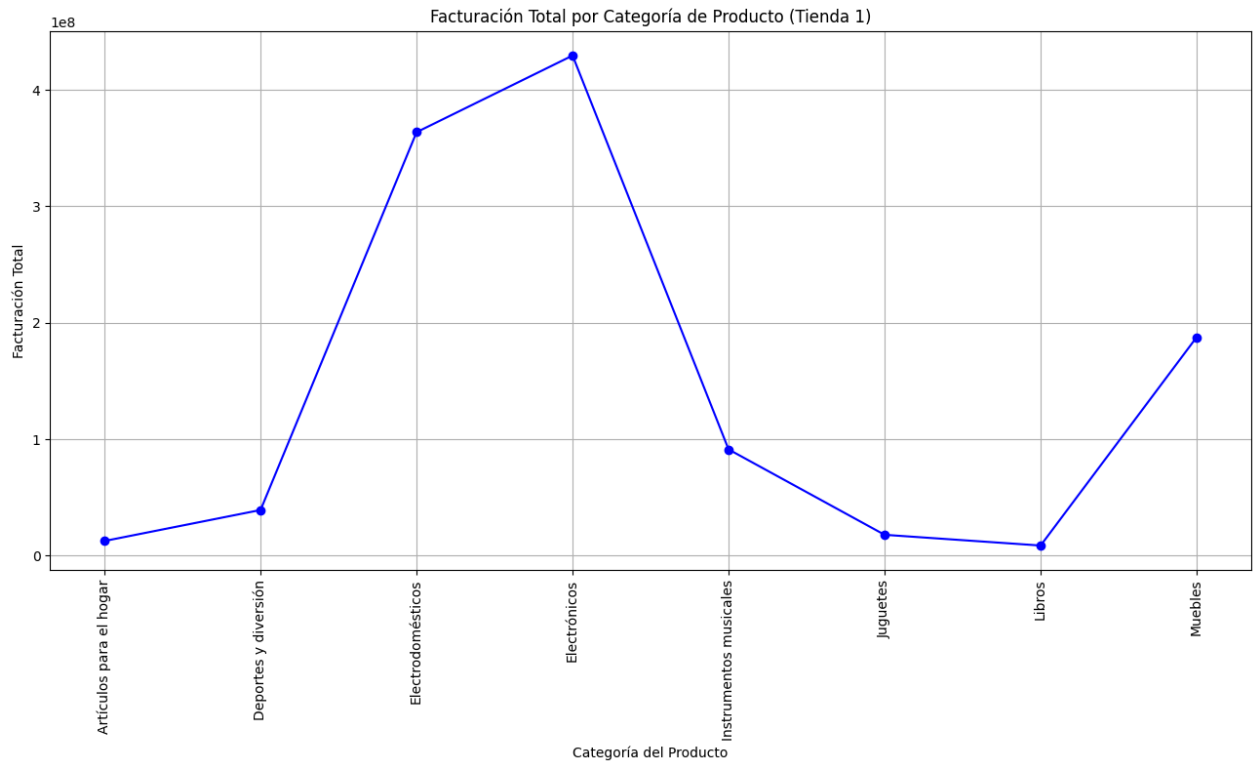
Distribución de Ventas por Categoría en Tienda 4

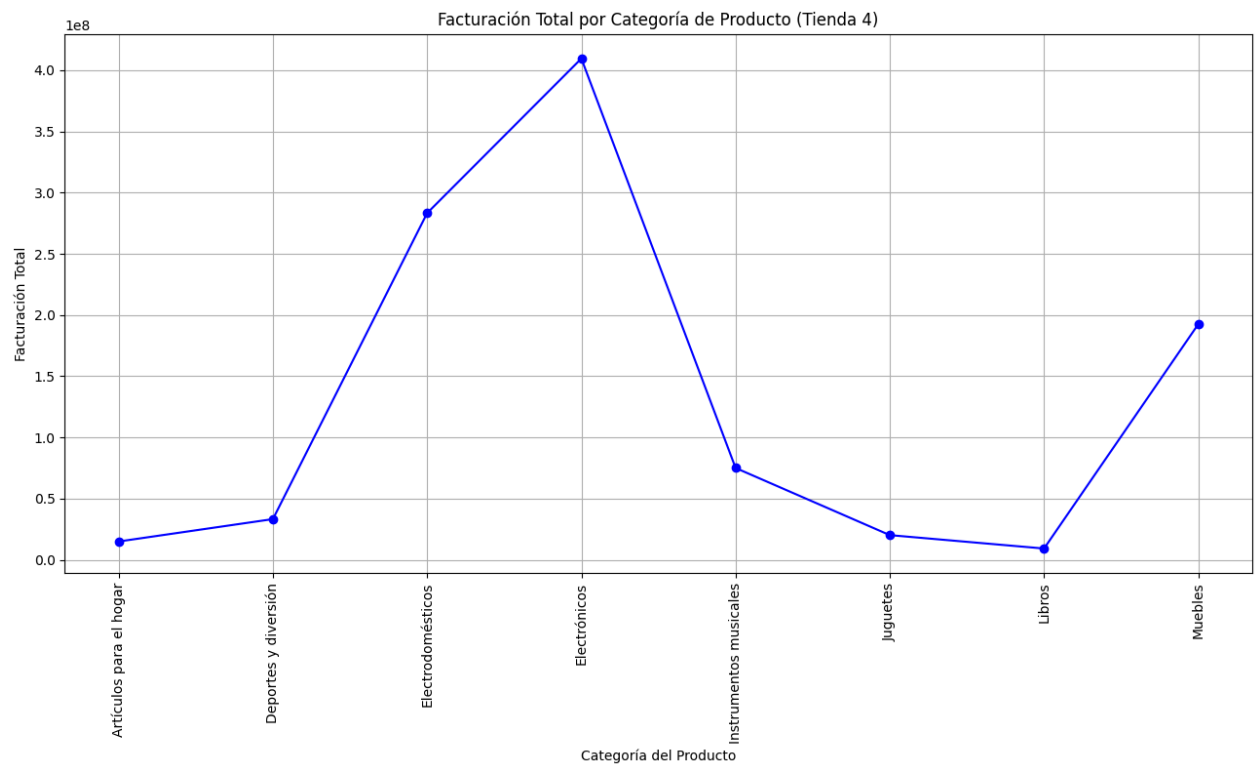
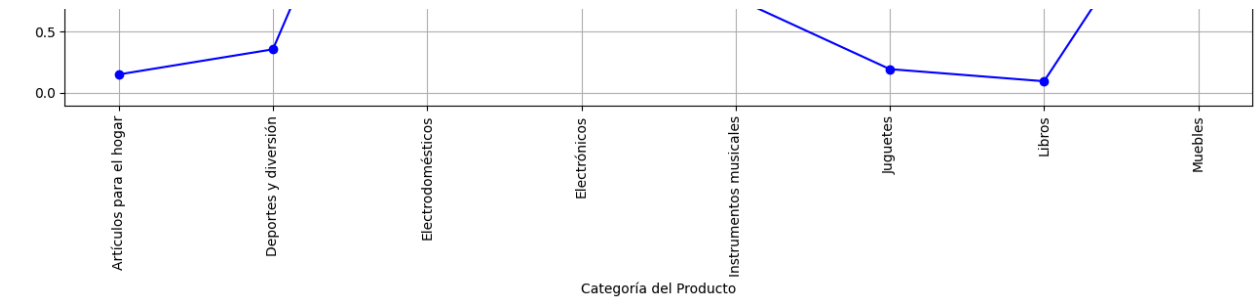



```
import matplotlib.pyplot as plt
def create_line_plot_category_sales(df, store_name):
    # Agrupar por categoría y sumar los precios (facturación)
    category_revenue = df.groupby('Categoría del Producto')['Precio'].sum()

    plt.figure(figsize=(13, 8))
    plt.plot(category_revenue.index, category_revenue.values, marker='o', linestyle='-',
    plt.xlabel('Categoría del Producto')
    plt.ylabel('Facturación Total')
    plt.title(f'Facturación Total por Categoría de Producto ({store_name})')
    plt.xticks(rotation=90)
    plt.grid(True)
    plt.tight_layout()
    plt.show()
    print("-" * 30)

# Gráficos de línea para la facturación por categoría en cada tienda
create_line_plot_category_sales(tienda, 'Tienda 1')
create_line_plot_category_sales(tienda2, 'Tienda 2')
create_line_plot_category_sales(tienda3, 'Tienda 3')
create_line_plot_category_sales(tienda4, 'Tienda 4')
```





✓ 3. Calificación promedio de la tienda

```
import matplotlib.pyplot as plt
# Calcular la calificación promedio para cada tienda
calificacion_promedio_tienda1 = tienda['Calificación'].mean()
calificacion_promedio_tienda2 = tienda2['Calificación'].mean()
calificacion_promedio_tienda3 = tienda3['Calificación'].mean()
calificacion_promedio_tienda4 = tienda4['Calificación'].mean()

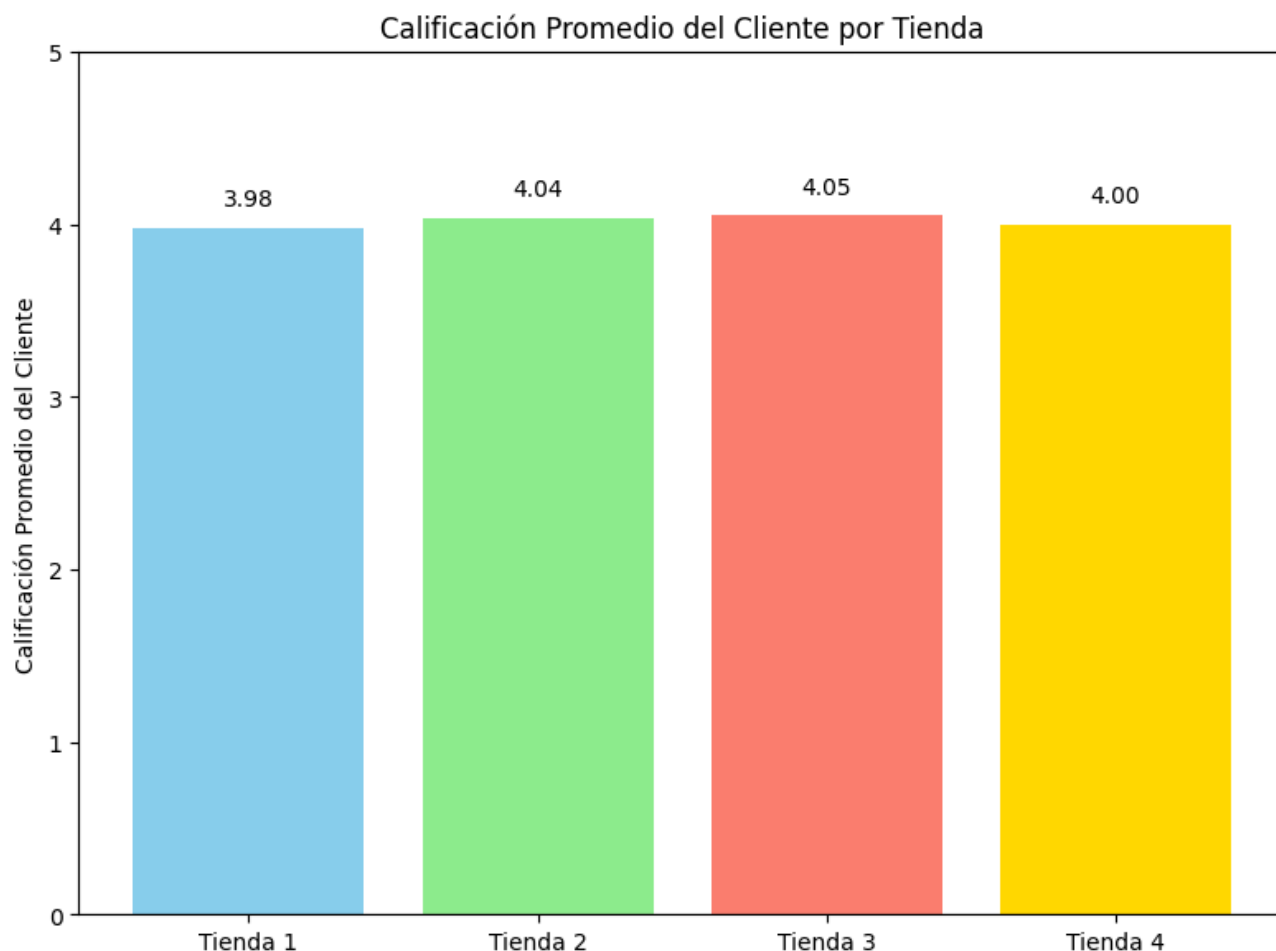
# Crear una lista con las calificaciones promedio y los nombres de las tiendas
calificaciones_promedio = [
    calificacion_promedio_tienda1,
    calificacion_promedio_tienda2,
    calificacion_promedio_tienda3,
    calificacion_promedio_tienda4
]
nombres_tiendas = ['Tienda 1', 'Tienda 2', 'Tienda 3', 'Tienda 4']

# Crear el gráfico de barras
fig, ax = plt.subplots(figsize=(8, 6))
ax.bar(nombres_tiendas, calificaciones_promedio, color=['skyblue', 'lightgreen', 'salmon'])

# Añadir etiquetas y título
ax.set_ylabel('Calificación Promedio del Cliente')
ax.set_title('Calificación Promedio del Cliente por Tienda')
ax.set_ylim(0, 5) # Establecer el límite del eje y entre 0 y 5 (rango de calificación)

# Mostrar los valores exactos sobre cada barra
for i, v in enumerate(calificaciones_promedio):
    ax.text(i, v + 0.1, f"{v:.2f}", ha='center', va='bottom')

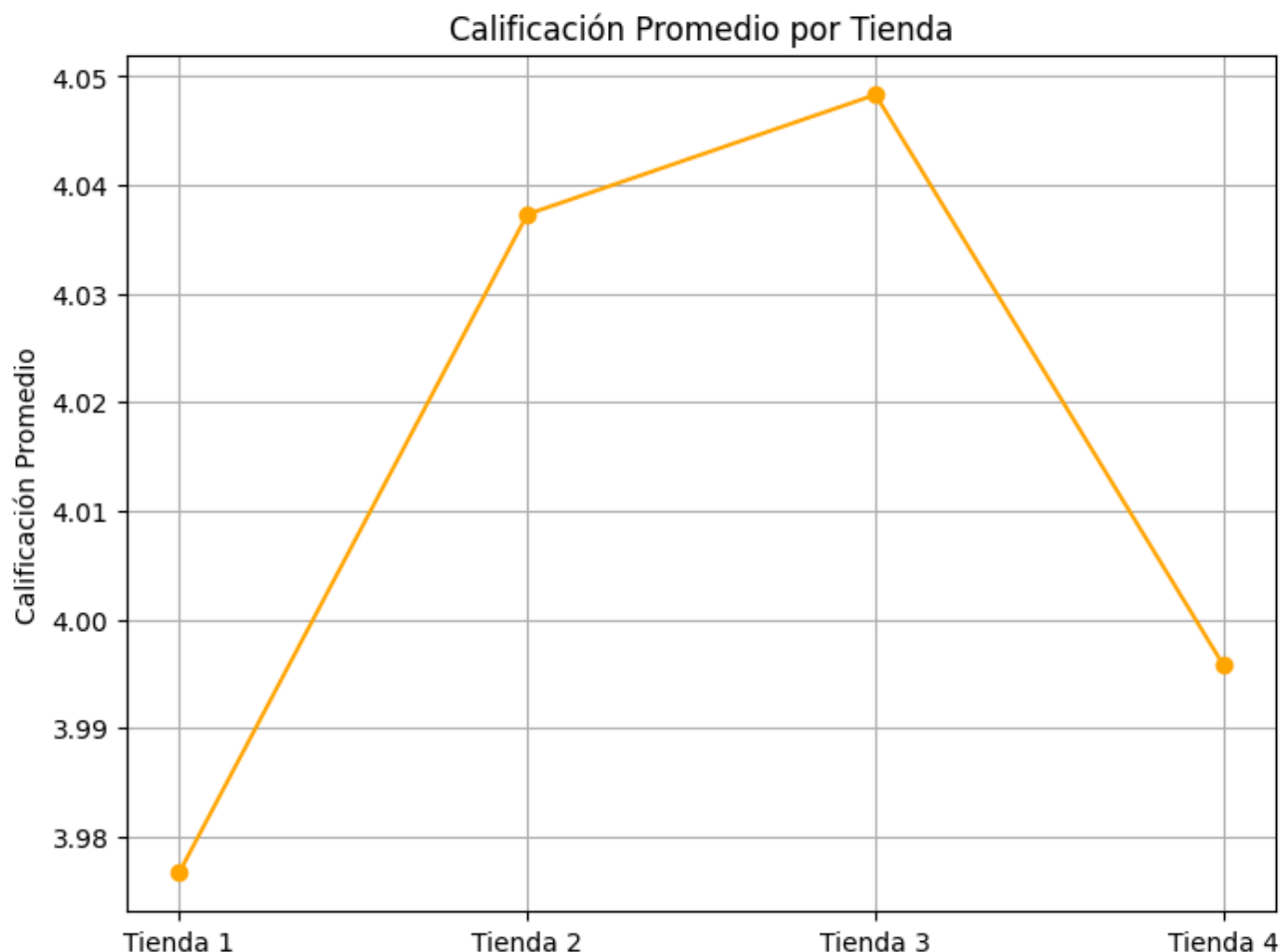
plt.tight_layout()
plt.show()
```



```
import matplotlib.pyplot as plt
calificacion_tienda_1 = tienda['Calificación'].mean()
calificacion_tienda_2 = tienda2['Calificación'].mean()
calificacion_tienda_3 = tienda3['Calificación'].mean()
calificacion_tienda_4 = tienda4['Calificación'].mean()

calificaciones_promedio = [calificacion_tienda_1, calificacion_tienda_2, calificacion_tienda_3, calificacion_tienda_4]
tiendas = ['Tienda 1', 'Tienda 2', 'Tienda 3', 'Tienda 4']

plt.figure(figsize=(8, 6))
plt.plot(tiendas, calificaciones_promedio, marker='o', linestyle='-', color='orange')
plt.ylabel('Calificación Promedio')
plt.title('Calificación Promedio por Tienda')
plt.grid(True)
plt.show()
```



```
import pandas as pd
import matplotlib.pyplot as plt
# Combinar los dataframes para facilitar el análisis conjunto
all_stores = pd.concat([
    tienda.assign(Tienda='Tienda 1'),
    tienda2.assign(Tienda='Tienda 2'),
    tienda3.assign(Tienda='Tienda 3'),
    tienda4.assign(Tienda='Tienda 4')
])

# Calcular la calificación promedio por tienda y producto
satisfaccion_cliente = all_stores.groupby(['Tienda', 'Categoría del Producto'])['Califica

# Pivotear la tabla para tener las tiendas como columnas
satisfaccion_pivot = satisfaccion_cliente.pivot(index='Categoría del Producto', columns='

# Rellenar los valores NaN con 0 o un valor apropiado si un producto no se vendió en una
satisfaccion_pivot = satisfaccion_pivot.fillna(0) # 0 usa otro valor como la media genera

# Ordenar las categorías para una mejor visualización si es necesario
satisfaccion_pivot = satisfaccion_pivot.sort_index()

# Crear el gráfico de barras agrupadas
fig, ax = plt.subplots(figsize=(15, 8)) # Aumentar el tamaño del gráfico
```